Installation MEMO for SpringOS Version 1.2

K.Chinen and M.Misumi, StarBED project

2005/09/04

This memo describes installation steps of SpringOS and its information.

1 Overview

SpringOS is toolkit for network experiments. It setups nodes and network switches according to user specification. Moreover, it includes particular script language, *K* . The language supports waking programs on experiment entities and communication between entities. It makes scripting of experiment procedures/steps.

SpringOS ...

- Setups nodes
  - disks
  - IP address
- Setups network switches
  - VLANs
- Evaluates the language
  - waking programs upon experiment entity
  - communication between entities

is not...

- Multiple shell executor
- Traffic generator
- Layer 1 and/or 2 experiment toolkit

2 Terminology

experiment: a human activity using network equipments to design and analysis for network system.

node: a program excutable experiment entity, especially PC.

scenario: a sequence of experiment procedures/steps.

WoL: Wake on LAN; a PC booting method using NIC's BIOS.

VLAN: Virtual Local Area Network; is not Video LAN.

NIC: Network Interface Card;

IA: INTEL Architecture; i386 compatible CPU.

resource: actor hosts and IP addresses.

StarBED: The Internet simluator by 512 PCes and its building NICT founded.
Or the research project of Internet simulation/emulation using the facility.

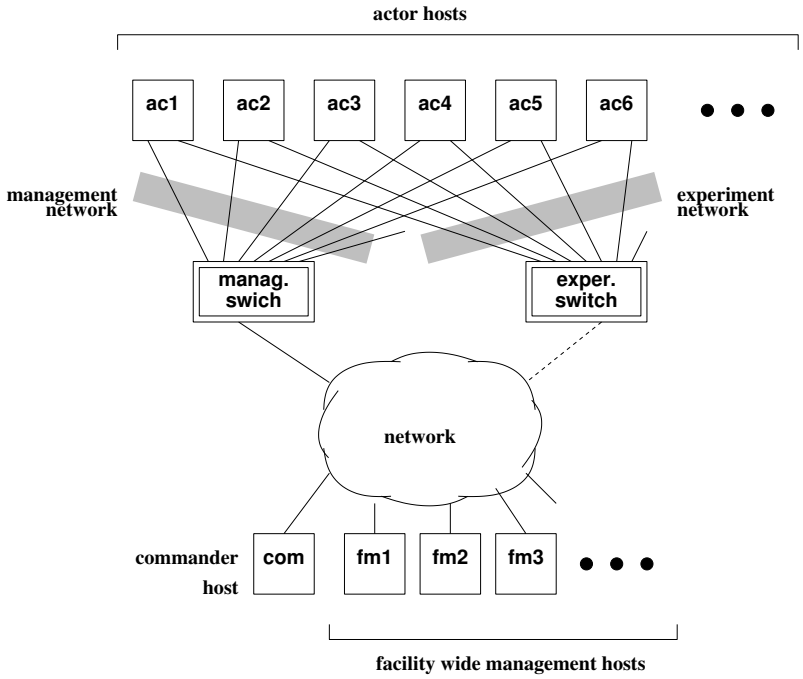

Figure 1: conceptual topology

3 Structure of ...

Hosts are divided 3 category.

- actor host (many)
  target programs for experiment run on them
- commander host (1)
  experiment driving program runs on the host
- facility side management host (1 and more)
  resource management, boot management

Programs are separated 2 region.

- facility wide management server programs
  - **`dhcpd`** (not included SpringOS)
  - **`tftpd`** (not included SpringOS)
  - **`ftpd`** (not included SpringOS)
  - **`fncp`**
  - **`erm`**
  - **`dman`**
  - **`wolagent`**
- experiment programs
  - driver
    - **`master`**
  - be driven
    - **`slave`**
    - **`ifscan`**
    - **`ni`**

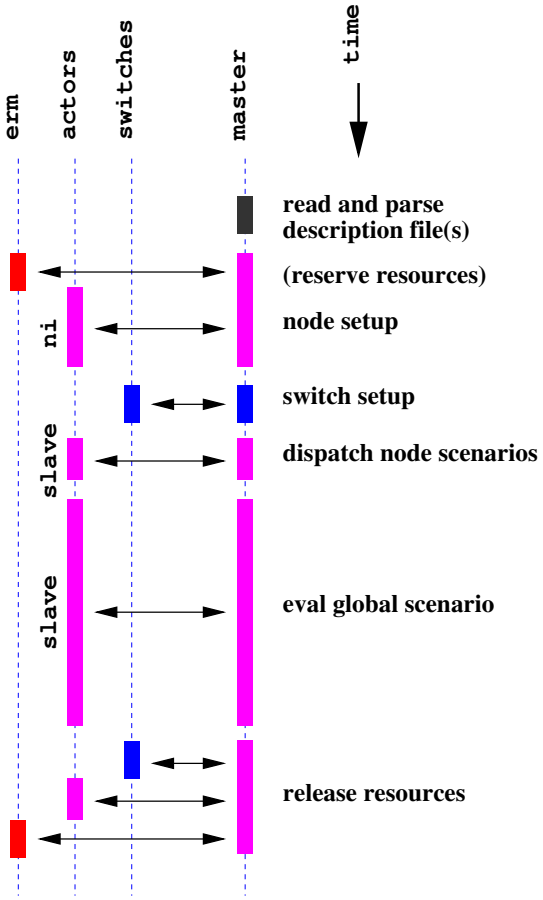

Figure 2: A Process flow of Experiment

4 Process Flows

Understanding of **master**'s behavior is important to use and debug SpringOS. Figure 2 depicts this process flow. In the beginning, **master** reads and parses experiment description file(s). Content of these files are shown later chapters.

After parsing file, the program enters communication of **erm** to decide actor host(s) as experiment node(s) and their spare(s). Spares are allocated to avoid insignificant errors. It will be discussed Section 14.1. According to result of **erm** communication, **master** trys to contact to actor hosts. When connection for **ni** is established, **master** issues OS installation command sequence. **master** waits until the number of nodes reaches to enough experiment running. Furthermore, timeout routine is ready to avoid forever waiting,

Switch(es) are setup after node setup. Because **master** knows the board and port number on switches which the host connected through **erm**'s database, the program can configure VLANs on switch(es). VLAN numbers are allocated from **erm**.

When nodes and switches are ready, it is time to run scenarios. **master** sends node scenarios to **slave**s along node definitions. It is dispatching roles to actor hosts.

At last, **master** release resources by communications.

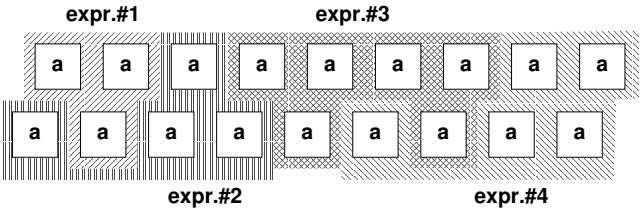

Figure 3: The test-bed is divided by multi experiments

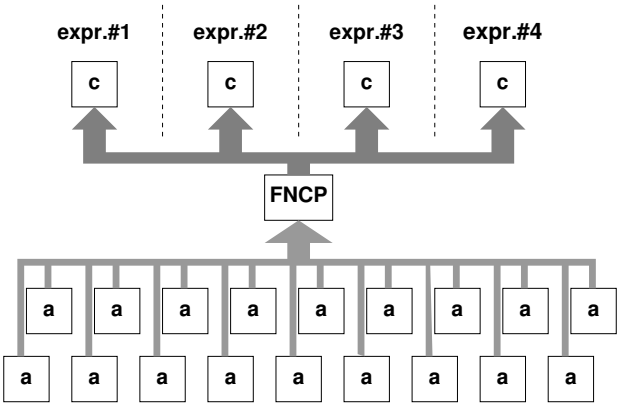

Figure 4: FNCP navigates actor hosts to commander host

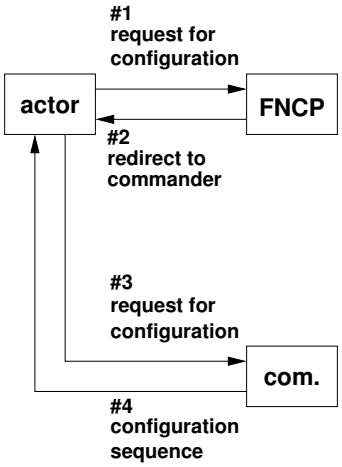

Figure 5: The navigation by HTTP redirection

4.1 The Navigation of Node Configuration

Since SpringOS have a purpose to support multi experiments in one test-bed (Figure 3), many experiment driver programs — `master` is one of them, can run on the test-bed. However, actor hosts don't know how many experiments are running and where are commander hosts of those in initially.

FNCP (facility node configuration pilot) have a role to solve the issue. `master` notifys a commander host for each actor host to FNCP. FNCP memorize those relations. Actor hosts can know commander host by asking to FNCP (Figure 4). The configuration of actor host is only to store FNCP's contact points. Other configuration (information about experiment, user, scenario, resources) are not required.

This navigation is a kind of routing. By HTTP redirection, FNCP notices a contact point to actor host. Figure 5 shows a sequence of its communication. So, a node configuration program has to communicate server at least twice.

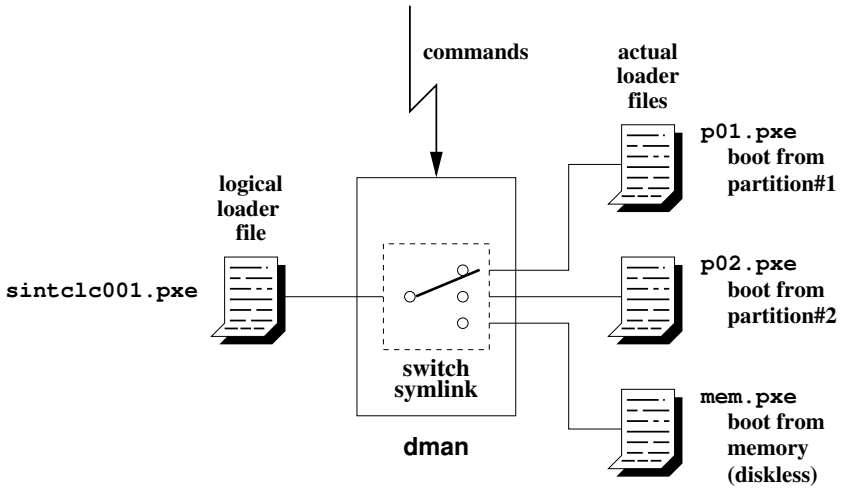

Figure 6: DMAN — symlink switcher

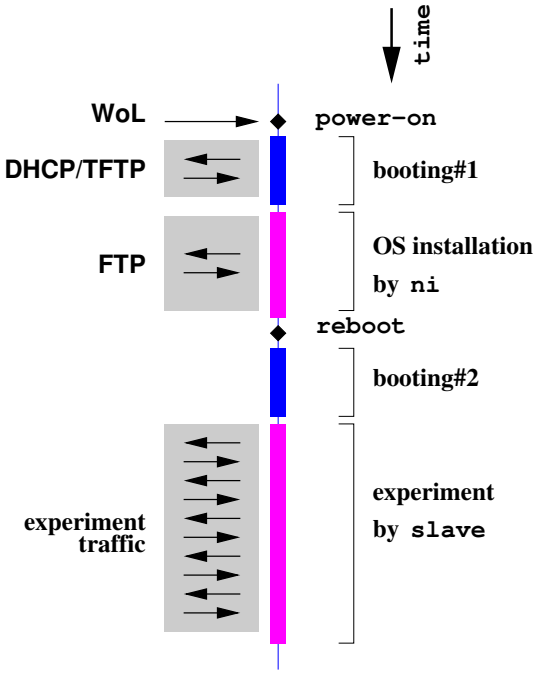

Figure 7: A Life of Actor Host

4.2 Loader Switching

SpringOS changes OSes on actor hosts by symlink (symbolic link) on the TFTP server directory. DMAN is an implementation of symlink switcher (Figure 6). By commands from foreign program, DMAN switches symlink for each actor host.

4.3 A Life of Actor Host

Figure 7 shows a typical life of actor host. Power-on by WoL magic packet. PXE booting by DHCP and TFTP. After **`ni`** waking, OS installation is started. **`ni`** reboot OS itself.

Installed OS boots in a moment. **`slave`** wakes and waits polling from **`master`**. To accept connection, **`master`** count this actor host as a candidate for experiment nodes. **`master`** sends a node scenario when the actor host are employed as node. Then, **`slave`** evaluates the scenario to drive experiment.

Table 1: Architecture/OS Availability

Category		i386 FreeBSD 4.2	i386 Turbo Linux7	SPARC Solaris 9
role	actor	√	√	
	commander	√	√	√
	manage		√	
program	dhcpd		√	
	tftpd		√	
	ftpd		√	√
	fncp		√	
	erm		√	√
	dman		√	
	wolagent		√	
	master/slave	√	√	√
	ifscan	√	√	N/A
	ni	√		

√: Available, N/A: Not Available, Blank: unknow

5 Required and Recommended

5.1 PCs

This section describe availability of PCs. Table 1 shows results of our test.

5.1.1 Actor Hosts

- IA base CPU
- UNIX link OS
- two and more NIC
  - PXE, WoL enable NIC at least one for booting
  - one and more for experiments

  We used following network interface card.

  - Intel Pro 100 (100Base-T)
  - Intel Pro 1000 (1000Base-T)
  - DEC related DE500 and its compatible (a.k.a ”Tulip”)
    There are many variation. We cannot promise you what are acceptable.
- compact chassis

  To gather a lot of PCs, compact chassis is recommended. rack-mount/blade type PC is more better.

5.1.2 Management Hosts

- architecture/OS

  No requirement, however UNIX like OS over IA base CPU is recommended. Because StarBED team never use other architecture/OSes (except **`ftpd`**, Solaris/SPARC and TurboLinux/i386 are used as **`ftpd`**) and this document expects them. You may use some architecture/OSes, probably. However this document can not cover them. please study your self.

- (**`dhcpd`**, **`wolagent`**) connected management network of actor hosts in L2 (data-link) view.

  DHCP(a kind of BOOTP) and WoL packets only flow on data-link layer. You should put several hosts when you use several hundreds actor hosts.

- (**`tftpd`**, **`ftpd`**) large storage

  diskimages often reach many tens GB. at least 40GB required, 160GB and more recommended.

- bootloader

  SpringOS expects filenames of bootloader. Disk booting named **`boot_p`** and digit (c.f., boot_p1 and boot_p5 .) See B.5.

5.1.3 Commander host

- Architecture/OS

  No requirement, however UNIX link OS over IA base CPU is recommended.

- POSIX thread library

  Master and slave are implemented as multi-threaded via POSIX Thread API. So, POSIX thread library is required. You may find them as **`libpthread.a`** in your system(s). Most experiment does not required serious scheduling. You can employ some compatible libraries, maybe.

- laptop/note PC

  Your daily used laptop/note PC is allowed. However, experiment driving program have to run and connect to entities during experiment. When you take long experiment, you may want to leave test-bed. Commander host cannot disconnect from test-bed in the case. Consider this point when you employ laptop PC as commander host.

5.2 Switches

SpringOS supports FastEthernet on CISCO switches with OS both IOS and CatOS. To avoid disturbance, you should reduce the connection to other switches. Isoratated switches are more better. However you have to keep connection(s) to control switches between the commander host and switches.

You have to check physical wire connection of switches to actor hosts. The result is used to make resource datafile of **`erm`** (Section 7.5.1).

5.3 Network topology

1) management network

   To management all PC, one NIC per PC connected this network. This network is managed with DHCP.

2) experiment network

   All NIC of each PCs except management network have to joind this network.

5.4 Password

SpringOS expects that switches are managed using common password.

6 Retrieve Files

Access WWW page **`http://www.starbed.org/softwares/ver-1-2/ver-1-2.html`**. Following files are most recently source code of SpringOS.

- erm-ver-1-2.tar.gz
- kuroyuri-ver-1-2.tar.gz
- ifsetup-ver-1-2.tar.gz
- fncp-ver-1-2.tar.gz
- dman-ver-1-2.tar.gz
- ni-ver-1-2.tar.gz
- snmpmine-20050802.tar.gz
- coil-20050902.tar.gz
- wolagent-20050904.tar.gz

Least version of them will be published. Please check **`http://www.starbed.org/softwares/`**

7 Management programs

7.1 dhcpd

dhcpd is not included SpringOS. You have to install that from other products. Some OSes includes that. This document expects TurboLinux built-in **dhcpd** (by ISC version 2.0pl5.)

Edit configuration file **dhcpd.conf**. It is located **/etc**, usually.

NOTE: This configuration makes binding among MAC, IP address and boot loader filename.

```
shared-network starbed-cde {
    server-identifier 172.16.3.101;

    option domain-name "si.star-bed.net";
    option domain-name-servers 172.16.3.101,172.16.1.101;
    option ip-forwarding off;
    option dhcp-server-identifier 172.16.3.101;
    use-host-decl-names on;

    # for Standerd Host
    group {
        next-server 172.16.3.101;
        option subnet-mask 255.255.254.0;
        option broadcast-address 172.16.3.255;
        option routers 172.16.3.254;

        subnet 172.16.2.0 netmask 255.255.254.0 {
        }

        host sintclc001 {
                hardware ethernet 00:00:4C:0F:77:C8;
                fixed-address 172.16.2.1;
                filename "sintclc001.pxe";
        }
        host sintclc002 {
                hardware ethernet 00:00:4C:0F:77:C6;
                fixed-address 172.16.2.2;
                filename "sintclc002.pxe";
        }
        host sintclc003 {
                hardware ethernet 00:00:4C:4F:A3:5E;
                fixed-address 172.16.2.3;
                filename "sintclc003.pxe";
        }
        host sintclc004 {
                hardware ethernet 00:00:4C:4F:A3:00;
                fixed-address 172.16.2.4;
                filename "sintclc004.pxe";
        }
        host sintclc005 {
                hardware ethernet 00:00:4C:4F:A3:26;
                fixed-address 172.16.2.5;
                filename "sintclc005.pxe";
        }
...
```

7.2 tftpd

tftpd is not included SpringOS. following document is described for tftp-hpa verion 0.28 .

7.3 ftpd

ftpd is not include SpringOS. Nothing condition is required to **ftpd**, mention above. You should care the direction of connection. SpringOS expects that the program supports PASS command (passive data connection.)

Check access permission of the program. Because experiment users have to write login account and password into the experiment description file, special user for SpringOS are recommended. The below example of description file expects special user 'install'. You should change that suits your situation.

7.4 fncp

fncp is a kind of HTTP server. Its features are so poor. The possibility of compiling error is low.

```
% gtar zxvf fncp-ver-1-2.tar.gz
% cd fncp/src
% ./configure
% make
```

Run **fncp** with no argument.

```
% ./fncp
```

7.5 erm

erm is the heart of SpringOS. This subsection describes compiling of that and its configuration.

```
% gtar zxvf erm-ver-1-2.tar.gz
% cd erm/src
% ./configure
% make
```

If your test-bed holds larger or equal 2048. You should change the maximum number of nodes. Edit ND_MAX_NODE in **Makefile**. Do not change **nd.h**. Following is an example to setup that 8192.

```
DEFS=-DND_MAX_NODE=8192
```

7.5.1 Resource Datafile

Next step, you have to make data files. **erm** required data file for resources and projects. Resource file (see **../data/starbed-resources**) consists definition of nodes (actor hosts, here) and VLANs like following:

```
vlan 800..831 JAIST-team

node sintcla001 [
    bootdisk,IDE
    net,manage,FastEthernet,00:00:4C:0F:76:74,,172.16.0.1,
    net,empty,FastEthernet,00:00:4C:0F:76:75,,,
    net,simulation,ATM,,"siatswa001,10",,
]

node sintclb023 [
    bootdisk,IDE
    net,manage,FastEthernet,00:00:4C:4F:A9:F8,,172.16.1.23,
    net,simulation,FastEthernet,00:00:4C:4F:A9:F9,"silaswa001,7/29",,
    net,simulation,ATM,,"siatswa004,132",,
]
```

Define VLAN with ID and description. Range is acceptable. **`800..831`** means range from 800 to 831. Some switches have some restriction for VLAN, reserved numbers or narrow range (e.g., 1024). See its documents.

`vlan` *VLAN-ID short-description*

Node definition consists with entity's name, disktype and NICs

---

`node` *node-name* [
  `bootdisk`, *disk-type*
  `net`, *network-type*, *media-type*, *MAC-addr*, *switch-port*, *IP-addr*,
]

---

node-name:

The name of node.

disk-type:

The type of boot disk.

literal	description
IDE	IDE; ATA
SCSI	SCSI

network-type:

The type for network.

literal	description
manage	management network
simulation	experiment network (historical reason)
empty	not used

media-type:

The type of media.

literal	description
ATM	Asynchronous Transfer Mode
Ethernet	Ethernet (10BASE); reserved, not used
FastEthernet	Fast Ethernet (100BASE)
GigabitEthernet	Gigabit Ethernet (1000BASE)
10GigabitEthernet	10 Gigabit Ethernet; reserved, not used

MAC-addr:

The MAC address of network :(colon) separated. Set empty in ATM, because it is undefined.

switch-port:

The pair of switch name and its port (with board if necessary) actor host connected. It is used to configure switch. You should describe under the manner of your switch (e.g., **`3`** or **`7/29`** .) Switch name is used for DNS resolving. Care the spell of that.

IP-addr:

The IP address of network .(dot) separated. Set this for management network. You should manage consitency between it and that of DHCP's. Otherwise, don't set this IP address. Because SpringOS allocates IP addresses automatic. When you set this, it will be conflict and/or confuse.

7.5.2 Project Datafile

Project datafile consists sets of tupple of user, password and project. Following example means user 'john', password 'ajapa' and project 'diskbench'. When you executes **master**, you have to give password as its argument. User and project name should be written into the experiment description file.

```
john:ajapa:diskbench
```

See sample file **../data/starbed-project** .

7.5.3 Execution of erm

Run **erm** with **-D** and **-U** options.

```
% ./erm -D resourefile -U projectfile
```

erm records ownershipment. The program read them at start time. So, it is "hot start". With **-C** option, you can "cold start" if necessary.

```
% ./erm -C -D datafile -U userfile
```

7.6 dman

dman is the symlink switcher. See also Section 4.2.

```
% gtar zxvf dman-ver-1-2.tar.gz
% cd dman/src
% ./configure
% make
```

Wake **dman** with **tftpd**'s direcotry (e.g., **/tftpboot** .)

```
% ./dman -D /tftpboot
```

7.7 coil

coil is a boot loader for multi-partition host. You can boot OS on the partition of actor hosts you wont by this program. Compile that using **make**.

```
% gtar xvfz coil.tar.gz
% cd coil/src
% make
```

After **make** you will get 6 boot loaders (**p1.bin** through **p6.bin**). These boot loaders kick the OS on the partition, respectively. You have to rename and copy them to **dman** expected (See B.5.)

```
% mv p1.bin boot_p1.bin
% mv p2.bin boot_p2.bin
% mv p3.bin boot_p3.bin
% mv p4.bin boot_p4.bin
% mv p5.bin boot_p5.bin
% mv p6.bin boot_p6.bin
% cp -p boot_p?.bin /tftpboot
```

When you want the boot loader for other partition (e.g., 10th partition), you can get that by rewriting one byte on these file. See **pch.pl** in this package.

7.7.1 To backup MBR

Coil is not aim to use as MBR (Master boot record; the first sector of the disk). However, It is necessary to take the backup of MBR sometimes. In some case, the system requires backup with a lot of sector to recover that. You should research your system.

The program **dd** is useful to backup MBR.

```
% dd bs=512 count=1 if=/dev/hda of=hda.mbr
```

To restore, run **dd**, too.

```
% dd bs=512 count=1 if=hda.mbr of=/dev/hda
```

7.8 wolagent

wolagent is a Perl script. You should check running of **perl** before starting **wolagent** .

```
% perl -v
```

When you got the version of **perl**, you can use **perl**, certainly.

```
% perl ./wolagent
```

8 Experiment Driving Programs

8.1 master

Experiment driver programs are gathered into the package, *Kuroyuri* .

```
% gtar zxvf kuroyuri-ver-1-2.tar.gz
% cd kuroyuri/src
% ./configure
% make
```

In rare case, **make** meats error at **sbpsh** compiling.

1) readline

   configure searches library readline, automatically. When you want to skip readline, comment-out line about READLINE in **config.h** .

   ```
   /* #undef HAVE_READLINE_READLINE_H */
   ```

   Remove **-lreadline** of libraries (LIBS) in **Makefile**. For example, when you find following line.

   ```
   LIBS=-ltermcap -lz -lrt -lmalloc -lpthread -lnsl -lsocket
   -L/usr/local/lib -lreadline -lfl
   ```

   After removing **-lreadline**.

   ```
   LIBS=-ltermcap -lz -lrt -lmalloc -lpthread -lnsl -lsocket
    -L/usr/local/lib -lfl
   ```

2) curses

   Some implentation of readline requires curses library. Add **-lcurses** to LIBS in **Makefile**.

8.1.1 Execution of master

There is a lot of arguments and options for master. Here, several arguments are appeared. See other documents or output of **-h** options. Typical execution form is following example:

```
% ./master -p ajapa -s pipin foo.sc bar.sc junk.sc
```

8.2 slave

slave is included in kuroyuri package. It will be made after making of **master**. To achieve the automatic running of experiment, you have to install **slave** in every actor hosts and setup it as auto waking by **rc.local** or similar methods.

Execution of slave is very simple. With no argument, you can start this program.

```
% ./slave
```

When you do experiments with **skipnsetup** (Section 15), you should execute the program manually on every actor hosts.

8.3 ifscan

ifscan is includes in ifsetup package.

```
% gtar zxvf ifsetup-ver-1-2.tar.gz
% cd ifsetup/src
% make
```

To execute **ifscan** shows information of NICs.

```
% ./ifscan
# 3 interfaces; lo eth0 sit0
nic lo 00:00:00:00:00:00
nic eth0 00:0d:9d:db:a9:ed
nic sit0 00:00:00:00:00:00
```

9 Disk Handling Programs

Disk handling consists the program-**ni** and the diskimage to wake **ni**. This section show you how to install them.

Because **ni** writes disks, OS for **ni** should not access a target disk. A way to satisfy the requirement is diskless. You may employ other ways, using other disk (e.g, floppy, NFS). This document does not cover them.

9.1 ni

ni is the node initiator. The program reads and writes diskimage through network. So, it is means remote backup and restore.

By **configure** and **make**, you can get **ni**. Dynamic link library is not ready in the diskimage. You should use static link.

```
% gtar zxvf ni-ver-1-2.tar.gz
% cd ni/src
% ./configure
% make PDEFS=-static
```

To run **ni**, you have to know FNCP's service URL.

Try running of **ni** following command, manually. Replace IP address and port number to your **fncp**'s (default port number is 1238).

```
% ./ni http://1.2.3.4:1238/
```

9.2 SNMPmine

SNMPmine(**snmpmine**) is the SNMP responder. According to request, the program applys **shutdown** or **reboot** on actor host.

```
% gtar ztvf snmpmine-20050802.tar.gz
% cd mine/src
% ./configure
% make PDEFS=-static
```

The program aims at NEC MIB. Unfortunately, the MIB is not published.

9.3 Setup Diskless OS via PXE

The memo for making of diskless PicoBSD (a kind of FreeBSD) using PXE is published at StarBED WWW Server (**http://www.starbed.org/tips/netboot/index-j.html**.) Unfortunately, it is written by Japanese. This subsection is translated from the WWW page.

Steps in the subsection is required the PC which installed FreeBSD with kernel source. You have to pickup or install the PC. The author tested them on FreeBSD-4.6 .

9.3.1 pxeboot

pxeboot is bootloader for PXE. You can get that by making with LOADER_TFTP_SUPPORT.

```
# cd /usr/src/sys/boot/i386
# make -D LOADER_TFTP_SUPPORT
```

9.3.2 loader.rc

The file **loader.rc** is configuration file of PXE booting.

```
load /FreeBSD/kernel_${boot.netif.ip}.gz
load -t mfs_root /FreeBSD/diskimage_${boot.netif.ip}
autoboot 10
```

The variable **${boot.netif.ip}** is replaced to the IP address of each hosts. First line specifies kernel file (e.g., kernel_172.16.1.5.gz .) Second line specifies kernel file (e.g., diskimage_172.16.1.5.gz) also.

9.3.3 kernel

A MFS and MD_ROOT enabled kernel is required. You can use GENERIC kernel. After to apply **kgzip**, copy the kernel to the TFTP server.

9.3.4 diskimage

1) Login the host with root.

2) Download **http://www.starbed.org/tips/netboot/mkdiskimage-20050803.tgz** .

3) Extract files.

```
# tar xvfz mkdiskimage-20050803.tgz
# cd mkdiskimage-20050803
```

4) Edit **crunch.conf.local** if you want change contents of **crunch**.

```
# vi crunch.conf.local
```

5) Change the name and size of diskimage.

```
# vi mkdiskimage.sh
```

These parameters in following table are defined in the file **mkdiskimage.sh**. Without edit, the capacity of the diskimage is approx. 7MB.

name	description	default
IMAGE_SIZE	the size of diskimage	30000KB
IMAGE_NAME	the name of diskimage	diskimage

6) The directory **mfs_tree/etc** holds files in **/etc** of booted OS. Modify them if you want.

7) Make password database

Generate MD5 hashed string. Enter your password twice.

Underlined string '1zYB0U39b$kdL/yM3O7FaPNCvCsqYd3/' in following example is hashed string:

```
# openssl passwd -1
Password: your password
Verifying - Password: your password
$1$zYB0U39b$kdL/yM3O7FaPNCvCsqYd3/
```

Using **vipw**, add the hushed string into password file.

```
# cd mfs_tree/etc
# vipw -d ./
```

The password of root is empty in original.

```
root::0:0::0:0:root:/root:/bin/sh
```

Set hashed string as your password into the file.

```
root:$1$zYB0U39b$kdL/yM3O7FaPNCvCsqYd3/:0:0::0:0:root:/root:/bin/sh
```

8) Make the diskimage.

```
# cd ..
# ./mkdiskimage.sh all
```

9) Check existence the diskimage. If you find the diskimage (default name **diskimage**) in current directory, you got the diskimage.

9.4 Diskimage Customize for ni

For SpringOS, this section describes the customization of the diskimage.

9.4.1 Copy ni into Diskimage

Since the diskimage does not include **ni**, you should copy in to the file. The file is named 'ni02.fs' in next example. You will see that in Section 13.

```
# vnconfig -c -s labels vn0 $PWD/ni02.fs && mount /dev/vn0 /mnt
# cp -p /usr/home/ni/src/ni /mnt/usr/bin/ni
# cp -p /usr/home/mine/src/snmpmine /mnt/usr/bin/snmpmine
```

9.4.2 Automatic Waking of ni

You must configuration to wake **ni** in booting. Because actor hosts are rebooted automatically, SpringOS related programs expect automatic waking of **ni**.

To wake **ni**, you should setup booting procedures. Edit **rc.local**.

```
# vi /mnt/etc/rc.local
```

Typically command sequence for the waking is following:

```
/usr/bin/ni http://1.2.3.4:1238/
```

In actually, the **rc.local** in our **ni02.fs** have following commands:

```
ulimit -m 200000
ulimit -d 200000
echo -n "SNMPmine "
/usr/sbin/snmpmine > /dev/null 2>&1 &
echo -n "NI "
/usr/sbin/ni http://172.16.3.101:1238/nodeconfig.txt
```

ulimit, a shell built-in command, is a parameter modifier for user process. **snmpmine** is a self-destruction program to apply **reboot** and **shutdown**.

To generate and/or edit some script in **/etc** directory may makes simular behavior, also. This document does not cover that.

9.4.3 Deploy the Diskimage

Copy the diskimage file to **tftpd**'s data directory via **scp**, **ftp** and others.

```
# umount /mnt && vnconfig -u vn0
# scp -p $PWD/ni02.fs root@172.1.3.101:/tftpboot/FreeBSD
```

Table 2: Service Ports of SpringOS

program	port#	protocol	description
dhcpd	UDP 67	DHCP	server
	UDP 68	DHCP	client
tftpd	UDP 69	TFTP	control
ftpd	20	-	data (temporary)
	21	FTP	control
erm	1234	ERRP	resource information/reservation
dman	1236	DMAN	directory manipulation (symlink and information)
fncp	1238	HTTP	node configuration (redirect)
wolagent	5959	WOLAP	issue WoL magic packet
master	3456	HTTP	node configuration
	3458	ESQP	status reporting
slave	2345	*noname*	node scenario input
ni	80	HTTP	status reporting

10 Kick the Servers

Before starting server programs, you have to decide where does program run. See required and recommended conditions in previous Section 5.

Program dependency is following (make like syntax):

```
master: erm slave fncp dman dhcpd tftpd ftpd wolagent
salve: master ifscan fncp dman dhcpd tftpd ftpd
ni: master fncp dman dhcpd tftpd ftpd
ifscan:
dman: tftpd
erm:
fncp:
wolagent:
ftpd:
tftpd:
dhcpd:
```

Experiment programs (**master**, **slave** and **ni**) are depend servers deeply. Since **ifscan** is not used standalone, don't care that in this time.

Server programs are independent each other except **dman**. You can start them without order. Also you can start **dman** without order. However, its results are shown through **tftpd**. You should start **dman** after **tftpd** starting.

Furthermore, **dhcpd**, **tftp** and **ftpd** are configured as automatically starting at OS booting or on-demand starting by **inetd**(or **xinetd**) in many OSes. You should check the conflict between these server programs of SpringOS and those of OS built-ins.

The recommended order of SpringOS server programs booting is **erm**, **fncp**, **dman** and **wolagent**.

10.1 Check Service Ports

The number of listening ports is 13. Check these ports if you feel strange things or want to hack them. You may want them to other configuration (e.g., firewall).

Table 2 shows these ports. You can check them via **netstat** or **lsof**.

11 Manually Node Up/Down by sbpsh

The program-**sbpsh** in the package ”kuroyuri” is shell for the SpringOS environment. The program requires knowledge of environment parameters to you. You have to check those parameters to use the program. It is a good drill for to write the experiment description file.

```
% ./sbpsh
command>
```

11.1 Parameters

You can get parameters current state by ’set’ command.

```
command> set
trace off
prompt "command> "
user undef
project undef
rmpassword *none*
rm host localhost port 1234
snmport 161
tftpdman undef
nikernel ni-kernel
nidiskimage ni-diskimage
pxeloader recover_system/pxeboot
command>
```

When you want to know commands, type **help**.

```
command> help
commands:
    help
    sample
    poweron <hosts>
    reboot <hosts>
    poweroff <hosts>
    setdiskboot <hosts>
    setniboot <hosts>
    set snmpport <num>
    set rm <host> <port>
    set wolagent <host> <port> <ip-range>
    trace on|off
    quit

syntax:
    hosts ::= hosts,host                ex. c7,c10-12
    hosts ::= [a-b][0-9]+               ex. a23
            | [a-b][0-9]+-[0-9]         ex. c2-17
            | localhost
command>
```

Since **sbpsh** reads **.sbpshrc** before the command reception, you can set parameters automatically.

```
#
# .sbpshrc - sample for sbpsh
#
#trace on
set rm 172.16.3.101 1234
set tftpdman 172.16.3.101 1236
set wolagent 172.16.1.101 5959 172.16.1.0/23
set wolagent 172.16.3.101 5959 172.16.3.0/23
```

11.2 Manipulation

Using **setdiskboot**, you can change bootloader of actor host as disk booting. The second arguments **2** means second partition. You should set name of user and project before first command. And the program prompt to enter password. You have to enter password 'ajapa'.

```
command> set user john
command> set project diskbench
command> setdiskboot sintcle001 2
password:
```

You can register name of user and project in **.sbpshrc** . To protect privacy, the program rejects registration of password.

Commands **poweron**, **poweoff** and **reboot** apply them to actor hosts.

```
command> poweron sintcle001
```

The command **poweron** requires running **wolagent**. **poweroff** and **reboot** requires SNMP responder (e.g., SNMPmine) on actor host(s).

NOTE: You should apply **setdiskboot** before **poweron**, when you want to change OS.

Table 3: Options of pickup

option	description
-z	disable compression
-Q	disable autoreboot
-u *user*	user name
-p *passwd*	password
-j *project*	project name
-F *filename*	configuration file
-r *host:port*	RM's info
-f *host:port*	FNCP's info
-s *host:port*	ENCD's info
-k *host:port*	DMAN's info for tftpd
-P *path*	PXE boot loader
-K *diskimage:kernel*	NI's info
-X *str*	(pickup) prefix of destination (wipeout) diskimage

12 Manually Disk Backup/Restore

Disk backup and restore programs are named **pickup** and **wipeout**. These exist in the kuroyuri package.

12.1 The Parameter File

To avoid inheritance of bugs and simplify, these program do not support the language of **master**. They start with arguments and options. As time passes, they required many parameters. Command line operations may cause mistakes. Then, current versions support the file which describes parameters.

A traditional running example of **pickup**:

```
% ./pickup -r 172.16.3.101:1234 -k 172.16.3.101:1236 \
-f 172.16.3.101:1238 -s 172.16.220.118 \
-K FreeBSD/ni02.fs:recover_system/kernel_recover \
-X ftp://install:install@172.16.210.9/ sintclc004:2 | & tee p.log
```

In that case, you can use the parameter file **pickup.opt** like following:

```
-r 172.16.3.101:1234
-k 172.16.3.101:1236
-f 172.16.3.101:1238
-s 172.16.220.118
-K FreeBSD/ni02.fs:recover_system/kernel_recover
-X ftp://install:install@172.16.210.9/
```

Last running is equal to below. The option **-F** specify parameter file. The argument **sintclc004:2** means *2nd partition of host 'sintclc004'* .

```
% ./pickup -F pickup.opt sintclc004:2 | & tee p.log
```

Table 3 shows options of **pickup**. **wipeout** supports them also.

12.2 Backup – pickup

Using **-u**, **-p** and **-j**, set parameters for **erm** when you run current version **pickup**. These parameter should not be written into the parameter file. because they include privacy of user. You may omit **-X**, because it includes some privacy, also.

```
% ./pickup -F pickup.opt -u john -p ajapa -j diskbench\
-X ftp://install:install@172.16.210.9/ sintcle001:2 |& tee p.log
```

This example expects that user **install** is accessible 172.16.210.9 with password **install**.

12.2.1 The File Name Pattern of Backup

The backup-ed diskimage is named with hostname, partition and date. According to the FreeBSD manner, the program guesses device name by disk type. D and P are number of drive and partition. But D is always 0.

type	device name
IDE(ATA)	/dev/rad$D$sP
SCSI	/dev/rda$D$sP

Then, the program guesses filenames as following example:

disk type	argument	date	filename
IDE(ATA)	sintcle001:2	9 Aug 2004 13:17	sintcle001-rad0s2-200408091317.gz
SCSI	sintclc001:2	5 Jul 2004 11:33	sintclc001-rda0s2-200407051133.gz

The date part is generate by local time.

12.3 Restore – wipeout

The options of **wipeout** is same as that of **pickup** except **-X**. In **wipeout**, **-X** is the name of diskimage to restore. Moreover, you can specify multiple targets by listing them as arguments.

```
% ./wipeput -F pickup.opt -u john -p ajapa -j diskbench \
-X ftp://install:install@172.16.210.9/sintcle001-rad0s2-200408091317.gz \
sintcle001:2 sintcle002:2 sintcle003:2 |& tee w.log
```

13 The Facility Related Definition in the Experiment Description File

Experiment driving program, **master** runs according to the experiment description file. In the file, **master** knows test-bed situations. Following sample shows that.

```
#
# facility oriented information
#
# for StarBED 2003, 2004.
#
rmanager ipaddr "127.0.0.1" port "1234"
wolagent ipaddr "172.16.1.101" port "5959" ipaddrrange "172.16.1.0/23"
wolagent ipaddr "172.16.3.101" port "5959" ipaddrrange "172.16.3.0/23"
fncp ipaddr "172.16.3.101"
tftpdman ipaddr "172.16.3.101"

nidiskimage "FreeBSD/ni02.fs"
nikernel "recover_system/kernel_recover"
pxeloader "recover_system/pxeboot-nohang"
setuptimeout total 3600 warm 5

swtype name "silaswa001" type "IOS"
swtype name "silaswb001" type "CATOS"
swtype name "silaswb002" type "CATOS"
```

See syntax manual for furthermore information. Here, I describe short description. S means one definition is acceptable. don't define twice or more. M means multi definition are acceptable, if necessary.

item	#	description
rmanager	S	resource manager (erm)
wolagent	M	wolagent
fncp	S	facility node configuration pilot (fncp)
tftpdman	S	directory manipulator (dman) for TFTP
nidiskimage	S	file path of diskimage for ni
nikernel	S	file path of kernel for ni
pxeloader	S	file path of PXE loader
setuptimeout	S	node setup timeout
swtype	M	switches

You should seek data file of **erm** to describe switch's name.

14 Execution Test

To verify running, you should try experiment. Following listing is an example of experiment description file. 6 actor hosts are employed as 3 pair of **netperf**/**netserver**.

```
#
# you should read pre.sc before this file
#
# ./master pre.sc this.sc
#

user "starbed" "info@starbed.org"
project "starproj"
rbhfile "master.his"

encd ipaddr "172.16.220.118"
ipaddrrange "192.168.3.0/24"

nodeclass svclass {
  method "HDD"
  disktype "IDE"
  partition 2
  ostype "FreeBSD"
  diskimage \
    "ftp://install:install@172.16.210.9/sintclb001-rad0s2-200407061104.gz"
  netif media fastethernet
  scenario {
    netiffit "/tmp/ifscan"
    wakewait "/sbin/ifconfig" "/sbin/ifconfig" \
                self.netif[0].rname self.netif[0].ipaddr
    wakewait "/usr/bin/pkill" "pkill" "netserver"
    wake "/tmp/netserver" "/tmp/netserver"
    loop {
      recv x
      msgswitch x {
      "quit" {
        wakewait "/usr/bin/pkill" "pkill" "netserver"
        exit
        }
      }
    }
  }
}

nodeclass clclass {
  method "HDD"
  disktype "IDE"
  partition 2
  ostype "FreeBSD"
  diskimage \
    "ftp://install:install@172.16.210.9/sintclc001-rad0s2-200407051104.gz"
  netif media fastethernet
  scenario {
    netiffit "/tmp/ifscan"
    wakewait "/sbin/ifconfig" "/sbin/ifconfig" \
                self.netif[0].rname self.netif[0].ipaddr
    sleep 3
    recv dst
    sleep 3
    wakewait "/bin/ping" "/bin/ping" "-c" "10" dst
    wakewait "/tmp/netperf" "/tmp/netperf" "-H" dst
    send "cdone"
```

```
    sleep 3
  }
}

netclass ethclass {
  media fastethernet
}

nodeset client class clclass num 3
nodeset server class svclass num 3

netset ethnet class ethclass num 1
ethnet[0].ipaddrrange = "192.168.3.0/24"

attach server[0].netif[0] ethnet
attach client[0].netif[0] ethnet
attach server[1].netif[0] ethnet
attach client[1].netif[0] ethnet
attach server[2].netif[0] ethnet
attach client[2].netif[0] ethnet

scenario {
  sleep 60
  send client[0] haddr(server[0].netif[0].ipaddr)
  send client[1] haddr(server[1].netif[0].ipaddr)
  send client[2] haddr(server[2].netif[0].ipaddr)
  sync {
    msgmatch client[0] "cdone"
    msgmatch client[1] "cdone"
    msgmatch client[2] "cdone"
  }
  send server[0] "quit"
  send server[1] "quit"
  send server[2] "quit"
  sleep 10
  exit
}
```

You should replace server IP addresses, ports and filenames before running.

60 seconds sleeping of global scenario is buffer for switch setup. Because some switch takes several tens seconds to change port VLANs.

To apply facility definition file (here, I call that **pre.sc**) and this listing (**netperf.sc**) to **master** with password for **erm** and switches, the experiment will start.

```
% ./master -p rmpasswd -s swpasswd pre.sc netperf.sc
```

A lot of debugging options is available. Use them if you worry the behavior of the program. The option **-d** enables debug messages in all modules.

```
% ./master -t -d -p rmpasswd -s swpasswd pre.sc netperf.sc |& m.log
```

Using option **-U**, you can make the masking of debug messages. Following options order to record all messages expects module **eval** and **engen**.

```
% ./master -t -d -U eval,engen -p rmpasswd -s swpasswd \
pre.sc netperf.sc |& m.log
```

14.1 The Number of Spares

In actually, the number of actor hosts in the experiment is greater than user specified. To avoid error, **`master`** allocate spare host(s). The number of actor hosts are solved two parameters. These parameter is the ratio for spare and the minimum number of spare. Below equation express this procedure.

$$S = \sum_{i=1}^{m} \mathrm{MAX}\left(\left\lceil \frac{N_i \cdot r}{100} \right\rceil, N_i + k \right)$$

S	the total numeber of actor hosts
m	the number of nodesets
N_i	the number of defined node in i-th nodeset
r	the ratio for spare (default 105)
k	the minimum number of spare (default 1)

Therefore, eight actor hosts are allocated in last example. The number of spare actor hosts is two.

nodeset name	**user specified**	**allocated**	
		spare	**all**
`server`	3	1	4
`client`	3	1	4
total	6	2	8

Using **`sparenoderatio`** and **`sparenodemin`** in description file, you can change these parameters. When you set 100 and 0 to them, spare hosts are nothing. However, no spare means no error avoidance. be carefully.

```
sparenoderatio 100
sparenodemin 0
```

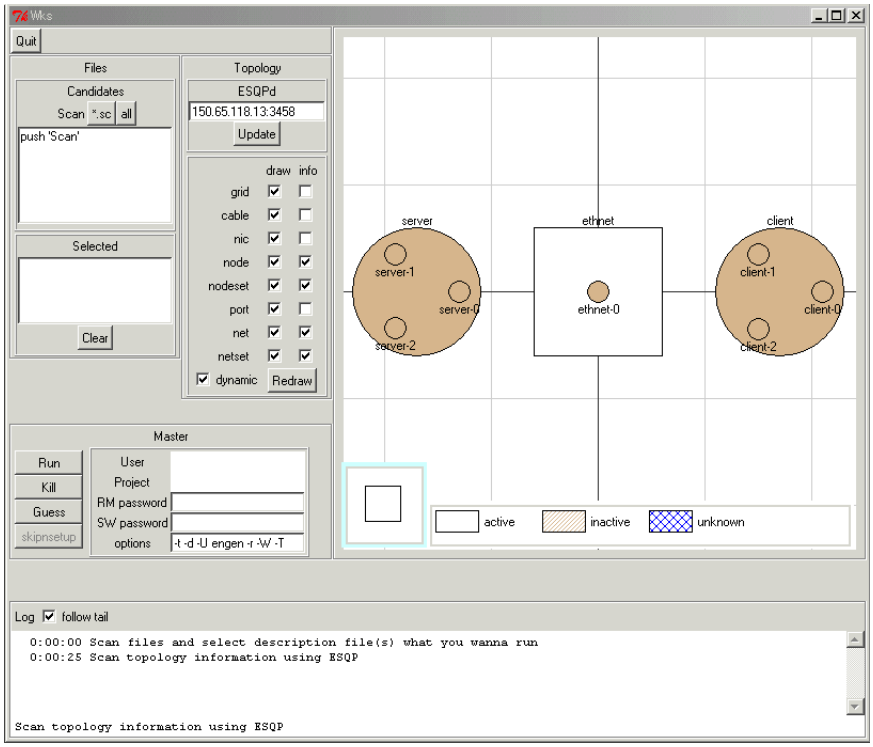

Figure 8: A Snap-shot of ks.pl

15 Execution with GUI

`ks.pl` is a wrapper of **`master`**. Figure 8 is a snap-shot of the program.

Certainly, the chance of **`ks.pl`** running is demonstration. Since disk operations take long time, you will want to skip that (Figure 9.) Such mode, skip disk operation is called ”skip node setup”(**`skipnsetup`**, in short) in SpringOS. It is originate in the perl script to enter such mode.

Following list shows steps to use **`ks.pl`**:

1) In ”Files” corner.

[1] Push ”*.sc”-button.

[2] Choice description file. The order of parsing is depend on the order of choice.

2) In ”Master” corner.

[1] Enter user and project.

[2] Enter passwords for RM (resource manager) and SW (network switches).

[3] Push ”Run”-button.

[4] Wait activation of ”skipnsetup” and drawing of topology.

[5] Push ”skinpsetup”

[6] See progress of experiments with animation

[7] (When ”@@END” message appears in log window, the experiment reaches the end.)

[8] Push ”Quit”

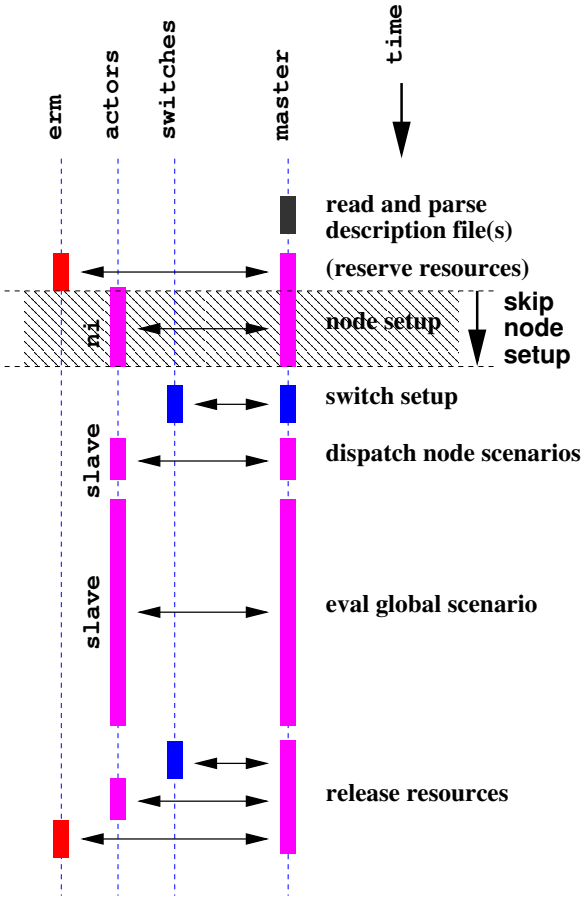

Figure 9: A Process Flow with skipnsetup

In case of the example at Section 14, the program runs with following results:

time segment	time[s]
pushing of ”Run” – activation of ”skipnsetup”	25
pushing of ”skipnsetup” – first annimation	49
pushing of ”skipnsetup” – second annimation	129
pushing of ”skipnsetup” – apparance ”@@END”	156

Initial options are ”-t -d -U engen -r -W -T”. It aim to run with **`skipnsetup`**. Change options if you want to enable disk operations and others.

16 Bugs and/or Drawbacks

UNIX specialized:

SpringOS focus UNIX and related. It do not care WINDOWS and other OS.

Poor capability for raw device name: SpringOS supports only single IDE or SCSI drive. Moreover, it supports partition 1,2 and 5. Device names are limited ”/dev/rad0sN”(IDE) and ”/dev/rad0sN”(SCSI).

Bootloader filenames:

SpringOS expects filename of bootloader for disk booting as ” boot_pN.bin ”. N is the partition number.

17 Contact Point

You can mail questions to ”info” at ”starbed” dot ”org”.

18 License

SpringOS is free. It has BSD like license. Further information see file in package.

References

SpringOS is designed for StarBED. You may find reasons for design of SpringOS.

[1] Toshiyuki Miyachi, Ken-ichi Chinen and Yoichi Shinoda: Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed, Tridentcom 2005, Trento, Italy, ISBN 0-7695-2219-X, pp.274–282, Feb, 2005.

[2] Homepage of StarBED Project. http://www.starbed.org/

[3] Tips of netboot (Japanese). http://www.starbed.org/tips/netboot/index-j.html

[4] Hokuriku IT Open Laboratory (Official Name of the Organization which holds StarBED) . http://www.hokuriku-it.nict.go.jp/english/

A Check List

Fill them of your testbed.

configuration items		values.....
dhcpd	IP address	
	port	
	pooled address	
tftpd	IP address	
	port	
	boot files directory	
dman	IP address	
	port	
	boot files directory	
ftpd	IP address	
	port	
	directory	
fncp	IP address	
	port	
	service URL	
erm	IP address	
	port	
wolagent	IP address	
	port	
master	IP address	
	port	

NOTE:

- You can use multiple **`ftpd`**.
- **`dhcpd`** and **`wolagent`** are required by each broadcast domain.

- Is fncp waked ? ___
- Is dhcpd waked ? ___
- Is tftpd waked ? ___
- Is dman waked ? ___
- Are actor hosts WoL bootable ? ___
- Do actor hosts supports PXE ? ___
- Is erm waked ? ___
- Do you check contents of database ? ___
  - How many hosts are defined ? _____
  - How many VLAN IDs are defined ? _____
  - Do you skip vendor's special VLAN ? ___
- Do you check contents of database, twice ? ___

B Historical Issues

B.1 SpringOS

The name 'SpringOS' originates from a spring in a bed. Spring supports the weight of object on a bed. Similarly, SpringOS supports a network test-bed.

B.2 sbrm

`erm` is called **`sbrm`** until early 2005. **`sbrm`** means "StarBED Resource Manager." The model of **`sbrm`** is useful in other test-bed. Then, StarBED team rename it as **`erm`**.

B.3 ifsetup

`ifsetup` is shell script. It is executor of **`ifconfig`**. After **`netif`** statement of the language is available, **`ifsetup`** is no longer.

B.4 wol_agent

`wolagent` was made Sep 2005 as an instead of **`wol_agent`**. Because rights of **`wol_agent`** are not clear.

B.5 Boot loaders

Initially, StarBED has the special boot loader for multi-partition (hereinafter also referrded to as "alice.") However, it is not free. Then StarBED project decided to develope free boot loader. It is named "coil." So, coil is substitute of alice.

Actually, alice is a set of boot loaders. Each boot loader boots the OS in the partition of which it takes charge. You have to know following 3 boot loaders.

name	partition	OS
boot_p1.bin	#1	WINDOWS Server 2000
boot_p2.bin	#2	FreeBSD 4.2
boot_p5.bin	#5 (first sub-partition on partition #4)	TurboLinux 7

B.6 'simulation' field in resource datafile of erm

Since StarBED is called "Internet Simulator" in early, experiment networks in the test-bed is called "simulation network". The term lefts in that file.

B.7 NEC MIB

NEC's management programs are installed in StarBED. To introduce the feature, SpringOS supports that. **`SNMPmine`** and **`sbpsh`** run under its manner.

B.8 SpringOS/VM

SpringOS/VM is a kind of extended SpringOS. It supports VMware. Users can increase the number of nodes for nextwork experiment. SpringOS/VM may be released if requested.

JAIST
JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
1990

StarBED Project

Contents